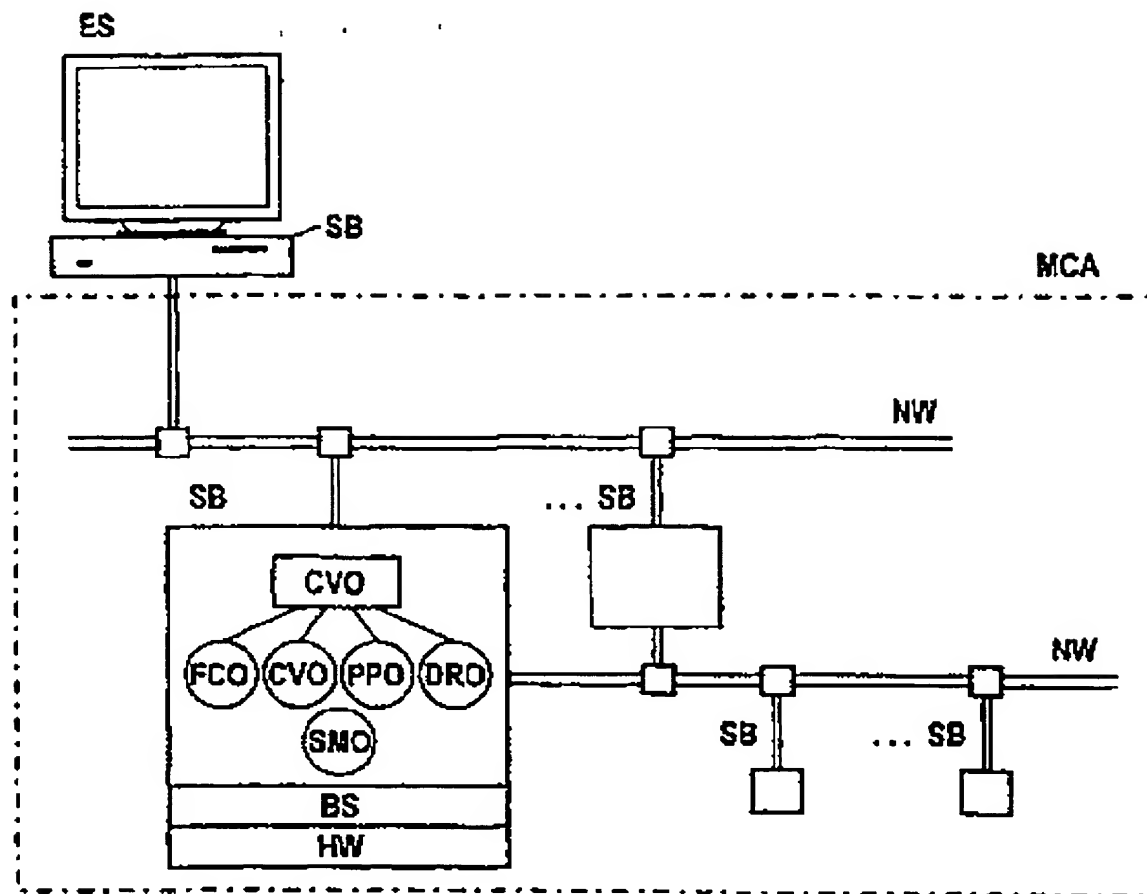


AN: PAT 2000-387898
TI: Process to project or program software application to control and monitor technical process uses instantiatable base object types with addressable interfaces to initialise
PN: WO200029917-A1
PD: 25.05.2000
AB: The instantiatable base objects include a program processing object type that processes a user-defined program and a driver object type to control the process hardware. The projection process involves the steps of selection and initializing the base object. The software application is performed on a runtime system and the projection on an engineering system. The programmer/projector selects base object types from a number of predefined base object types for the automation. The selection of a base object corresponds to an instantiation of the base object type. The existing instance of the base object type is a base object.; USE - To control movement of computer controlled production machinery. ADVANTAGE - Eases merging of control program and decreases number of central processing units needed.
PA: (SIEI) SIEMENS AG;
IN: HEBER T; HESS K; HORN W; KIRSTE S; KOSEL N;
FA: WO200029917-A1 25.05.2000; US6600964-B2 29.07.2003; **DE19853205-A1** 15.06.2000; EP1131686-A1 12.09.2001; US2001037161-A1 01.11.2001; CN1334933-A 06.02.2002; JP2002530730-W 17.09.2002; EP1131686-B1 23.04.2003; DE59905231-G 28.05.2003;
CO: AT; BE; CH; CN; CY; DE; DK; EP; ES; FI; FR; GB; GR; IE; IT; JP; LI; LU; MC; NL; PT; SE; US; WO;
DN: CN; JP; US;
DR: AT; BE; CH; CY; DE; DK; ES; FI; FR; GB; GR; IE; IT; LU; MC; NL; PT; SE; LI;
IC: G05B-015/02; G05B-019/042; G06F-009/44; G06F-019/00;
MC: T01-F06; T01-G05C; T01-J07B; T01-J20B; T06-A04B1; T06-A07A; T06-A08;
DC: T01; T06;
FN: 2000387898.gif
PR: DE1053205 18.11.1998;
FP: 25.05.2000
UP: 21.08.2003

BEST AVAILABLE COPY

THIS PAGE BLANK (USPTO)



THIS PAGE BLANK (USPTO)



Die vorliegende Erfindung betrifft ein Verfahren zur Steuerung technischer Prozesse, insbesondere zur Bewegungssteuerung von Produktionsmaschinen, wobei mittels einer zentralen Recheneinheit ein Steuerungsprogramm abgearbeitet wird, das auf einer Instanzierbarkeit sowie einer bedarfsgerechten Verschaltung von Softwarekomponenten mit vorgebbarer, zumindest parametrierbarer Funktionalität basiert.

Bisher bekannte Verfahren zur Automatisierung technischer Prozesse basieren im wesentlichen entweder auf einer "SPS-Funktionalität" oder auf einer "CNC-Funktionalität". Da im Rahmen solcher Funktionalitäten ein gewisser Funktionsumfang fest vorgegeben ist, ist eine optimale Anpassung an die Anforderungen eines speziellen Prozesses häufig nur bedingt möglich, wobei im konkreten Anwendungsfall oft eine ganze Gruppe von Funktionen überflüssig ist (z. B. beim Einsatz einer CNC-Steuerung, die eigentlich für Werkzeugmaschinen konzipiert ist, an Verpackungsmaschinen).

Des weiteren existieren im Hinblick auf den jeweiligen Anwendungsfall individuell ausgelegte Einzwecksteuerungen in Form speziell programmierter Steuerungssoftware.

Aus der DE 197 40 550 ist außerdem eine Vorrichtung zum Steuern eines technischen Prozesses und/oder zur Steuerung der Bewegung einer Verarbeitungsmaschine bekannt, die ein Steuerprogramm abarbeitet. Dieses Steuerprogramm besteht aus einer Vielzahl von Software-Modulen. Prozeßsteuerungsfunktionalitäten von an sich bekannten speicherprogrammierbaren Steuerungen und Bewegungsfunktionalitäten von an sich bekannten NC-Steuerungen sind in einem einheitlichen, konfigurierbaren Steuerungssystem verwirklicht. Die einzelnen Software-Module werden hier jedoch durch jeweils eine Teilsteuerung abgearbeitet, so daß für jedes Software-Modul eine zentrale Recheneinheit vorzusehen ist.

Aufgabe der vorliegenden Erfindung ist es, das Zusammenfügen des Steuerungsprogramms zu vereinfachen und die Anzahl der notwendigen zentralen Recheneinheiten zu verringern.

Diese Aufgabe wird mittels eines Verfahrens zur Projektierung bzw. Programmierung einer Softwareapplikation zur Steuerung und/oder Überwachung eines externen technischen Prozesses dadurch gelöst, daß instanzierbare Basisobjekttypen mit adressierbaren Schnittstellen zur Parametrierung und Verschaltung vorgesehen sind, daß als instanzierbare Basisobjekttypen zumindest ein Programmverarbeitungsobjekttyp zur Abarbeitung eines benutzerdefinierbaren Programms und zumindest ein Treiberobjekttyp zur Ansteuerung der Prozeßhardware vorgesehen ist und daß die Projektierung zumindest die Schritte der Selektion, Parametrierung und Verschaltung der Basisobjekte umfaßt.

Das Steuerungsprogramm besteht somit im wesentlichen aus Software-Objekten mit adressierbaren Schnittstellen. Zur Projektierung und Programmierung einer konkreten Softwareapplikation wählt der Projektierer/Programmierer aus einer Menge vordefinierter Basisobjekttypen die für das jeweilige Automatisierungsvorhaben erforderlichen Basisobjekttypen aus. Die Selektion eines Basisobjekttyps entspricht im Rahmen der Projektierung/Programmierung der Instanziierung des entsprechenden Basisobjekttyps. Die jeweilige Instanz eines Basisobjekttyps ist ein Basisobjekt. Mittels der Schnittstellen der Basisobjekte sind die Basisobjekte parametrier- und verschaltbar, so daß sich die jeweils entsprechend der konkreten Anforderungen ausgewählten Basisobjekte über ihre Schnittstellen zu einem Steuerungsprogramm zusammenzufügen lassen und somit schließlich

eine Softwareapplikation für konkrete Steuerungsfunktionen bilden.

Als Basisobjekttypen kommen Objekttypen mit grundsätzlich vorgegebener Funktionalität in Frage. Ein Programmverarbeitungsobjekt ermöglicht die Ausführung einer konkreten durch den jeweiligen Benutzer individuell für jedes Programmverarbeitungsobjekt vorgebbaren Anwendungsfolge.

Zur Ansteuerung der Prozeßhardware sind die Treiberobjekte vorgesehen, wobei für unterschiedliche Hardwarekomponenten jeweils unterschiedliche Treiberobjekte vorgesehen sind, auch wenn in der nachfolgenden Beschreibung verkürzt nur noch von Treiberobjekten gesprochen wird.

Vorteilhaft kann zur Vorgabe einer Führungsgröße für eine Hardware-Teilkomponente des technischen Prozesses ist als instanzierbarer Basisobjekttyp ferner ein Führungsgrößenobjekttyp vorgesehen sein.

Darüber hinaus oder alternativ kann zur Regelung einer Hardware-Teilkomponente des externen technischen Prozesses als instanzierbarer Basisobjekttyp ein Reglerobjekte vorgesehen sein.

Vorteilhafterweise erfolgt die Projektierung/Programmierung auf einem Engineering-System, während die Softwareapplikation auf einem Runtime-System zur Ausführung kommt. Die Projektierung bzw. Programmierung ist damit unabhängig von der Ausführung der jeweiligen konkreten Software-Applikation.

Die Erfindung erfaßt nicht nur die Projektierung/Programmierung einer Software-Applikation sondern auch die jeweilige gemäß der Erfindung projektierte/programmierte konkrete Software-Applikation, insofern als die Steuerung und/oder Überwachung des technischen Prozesses mittels einer gemäß dem erfindungsgemäßen Verfahren projektierten bzw. programmierten Software-Applikation erfolgt. Die Software-Applikation zeichnet sich durch die jeweils verwendeten Basisobjekte und deren konkrete Verschaltung aus.

Weitere Merkmale, Vorteile und Anwendungsmöglichkeiten der Erfindung ergeben sich aus den Unteransprüchen, der nachfolgenden Beschreibung eines Ausführungsbeispiels anhand der Figuren und der Figuren selbst. Dabei bilden alle beschriebenen und/oder bildlich dargestellten Merkmale für sich oder in beliebiger Kombination den Gegenstand der vorliegenden Erfindung, unabhängig von ihrer Zusammenfassung in den Ansprüchen oder deren Rückbeziehung.

Dabei zeigt

Fig. 1 eine exemplarische Struktur einer Bewegungssteuerungsapplikation (MC-Applikation),

Fig. 2 eine Detailansicht einer Komponente einer MC-Applikation mit einem Programmverarbeitungsobjekt und einer Hardwareanbindung über entsprechende Treiberobjekte und

Fig. 3 eine schematische Darstellung der Entwicklungsumgebung zur Projektierung bzw. Programmierung der MC-Applikation auf einem Engineering-System einerseits und zum Ablauf der projektierten bzw. programmierten MC-Applikation auf einem Runtime-System andererseits,

Fig. 4 ein Objekt-Modell,

Fig. 5 Interface-Definitionen und

Fig. 6 eine Objektstruktur einer Schlauchbeutelmaschine.

Jedes Automatisierungsvorhaben setzt auf Seiten des zu automatisierenden Prozesses auf einem Sortiment von Hardwarebaugruppen auf. Im Falle eines Bewegungssteuerungssystems, das im Rahmen der nachfolgenden Beschreibung vordringlich betrachtet werden soll, sind dies z. B. Einachs-, Dreiachs-, Vierachsbaugruppen mit entsprechenden Antriebs-

ben.

Auf der Software-Seite ist zur Handhabung der üblicherweise verarbeitungs-asynchron zu berücksichtigenden Ereignisse des technischen Prozesses ein Echtzeitbetriebssystem und darüber hinaus ein Netzwerk zum Aufbau eines dezentralen Steuerungssystems mit verteilter Steuerungsfunktionalität erforderlich.

Das in Fig. 1 dargestellte Bewegungssteuerungssoftwaresystem, im folgenden MC-Softwaresystem, umfaßt zumindest ein Engineeringssystem ES (Off-Line) und ein Runtime-system RS (On-Line).

Das ES dient dabei dem Anwender zur Erstellung einer Bewegungssteuerungsapplikation, im folgenden MC-Applikation_MCA genannt. Das RS übernimmt deren Ausführung.

Zur Erstellung einer MCA gehören die Systemkonfiguration, die Erstellung der Anwenderprogramme auf Hochsprachenniveau und die Überführung dieser Informationen in eine RS-intern ausführbare Form (Executable). Das RS dient der Ausführung des Executables.

Jede in Rahmen des RT zum Einsatz kommende Hardwarekomponente wird als Systembaustein SB bezeichnet. Es kann aus einem oder mehreren, über ein Netzwerk NW verbundenen SB's bestehen. SB's sind dabei Prozessrechner mit dedizierter, auf die Steuerungsfunktionalität zugeschnittener Peripherie (auch PC möglich), auf denen notwendige und optionale RS-Komponenten ablaufen.

Das Engineeringssystem ES läuft auf einem Programmiergerät oder Personal Computer unter einem Standardbetriebssystem ab. Das ES besitzt für Inbetriebnahme- bzw. Testzwecke dieses Executables einen Zugang zum RS.

Die konkrete Steuerungsstruktur entsteht durch Verschaltung von auf den SB's, gemäß Fig. 1, instanziierten sogenannten Basisobjekten BO, die mit unterschiedlicher Funktionalität ausgestattet sind. Je nach Anwendung können die Basisobjekte BO auf SB' verteilt (instanziiert) werden. Des weiteren weist zum Zugriff auf die Standardhardware HW grundsätzlich jeder Systembaustein SB ein Betriebssystem BS auf.

Beispiele für Basisobjekte BO sind:

- Reglerobjekte FCO für verschiedene Reglerfunktionen,
- Führungsgrößenobjekte CVO als Führungsgrößen-erzeuger, die vor allem Bewegungsbefehle ausführen,
- Programmverarbeitungsobjekte PPO zur Abarbeitung vorgegebener Anwenderprogramme,
- Treiberobjekte DRO zur Anpassung der Steuerung an Schnittstellen der verschiedene Hardwarebaugruppen sowie
- Systemmanagerobjekte SMO als Schnittstelle zum Betriebssystem,
- Boot-Objekt BOO zur Speicherung des Executables der MCA und zur Organisation des Systemanlaufs.

Alle Basisobjekte BO außer den Systemmanagerobjekten SMO und einigen Treiberobjekten DRO sind von "außen" - also durch den Projektierer oder Entwickler - instanziiierbar. Die Basisobjekte BO selbst existieren in der Steuerung als Firmware des RS.

Bei den Basisobjekten BO wird unterschieden zwischen anwenderkonfigurierbaren Funktionalobjekten und notwendigen Systemobjekten. Anwenderkonfigurierbare Funktionalobjekte sind z. B. die Reglerobjekte FCO, die Führungsgrößenobjekte CVO, die Programmverarbeitungsobjekte PPO, einige Treiberobjekte DRO und ein RS-weites exklusives Boot-Objekt BOO. Notwendige Systemobjekte sind dagegen die Systemmanagerobjekte SMO und spezielle Trei-

berobjekte DRO.

Funktionalobjekte FCO, CVO und PPO sind typische Objekte zur Projektierung einer Steuerungslösung nach den Anforderungen der jeweiligen Anwendung. Notwendige Systemobjekte SMO, DRO werden zum Netzwerkzugang auf einem Systembaustein SB bei Anlauf der Firmware angelegt und verschaltet und sind von außen nicht instanziiierbar. Damit ist der Zugang auf jede Hardware HW gewährleistet.

Eine MC-Applikation MCA wird durch Verschaltung, Konfiguration und Anwenderprogrammierung (bei PPO's) von Basisobjekten erreicht. Vorkonfigurierte BO-Gruppen können im ES gebildet und/oder verarbeitet werden. Damit erfolgt eine Makro-Bildung in Form von Applikationsmodulen SAM auf Anwenderebene.

Die Verschaltung der Basisobjekte BO wird grundsätzlich von "außen" mittels des ES konfiguriert und im Executeable abgelegt. Die Herstellung der realen Verschaltung erfolgt auf Grund dieser, im Executeable enthaltenen Informationen durch das BOO oder das ES.

Die Verschaltung der Basisobjekte BO erfolgt über dafür vorgesehene Schnittstellen an den BO's. Diese Schnittstellen sind im RS und ES eindeutig adressierbar. Es wird zwischen optionalen und nichtoptionalen Schnittstellen unterschieden. Nichtoptionale Schnittstellen dienen allgemeinen Systemfunktionen wie dem Empfang von Konfigurationsdaten, dem Variablen austausch oder für Debug- und Trace-Zwecke. Optionale Schnittstellen repräsentieren funktionale Spezifika eines BO's nach außen.

Das ES benutzt zur Repräsentation von BO-Typen bzw. -Instanzen im RS Stellvertreterobjekte bzw. Instanzen als duale Darstellung. Über diese Stellvertreterobjekte erhält das ES Informationen über die Eigenschaften von BO's und (bei aktivem RS-Zugang) Zugang zu den BO's und deren Schnittstellen im RS. Diese Beziehungen sind in Fig. 4 dargestellt.

Schnittstellen im Engineeringssystem (ES-Interfaces)

Alle Basisobjektclassen verfügen als nicht optionale ES-Interfaces über Kommando- und Konfigurationsschnittstellen, wobei bei Bedarf eine Trennung erfolgen kann. Weitere optionale ES-Interfaces sind Variablen-Schnittstellen, Debug-Schnittstellen bzw. Trace-Schnittstellen.

In Fig. 5 sind Objektschnittstellen beschrieben, wobei zur Verknüpfung von zwei Schnittstellen folgende Matching-Regeln (alle AND-verknüpft) verwendet werden: Verknüpft wird wenn:

1. die Interface'd's gleich sind
2. die Interface-Direction komplementär zueinander sind
3. der Transmit-Type gleich ist
4. die Connectivity dies noch zuläßt.

Schnittstellen im Laufzeitsystem RS (RS-Interfaces)

Die Festlegung der RS-Interfaces erfolgt im Bewegungssteuerungslaufzeitsystem direkt. Die Schnittstellen werden im Engineeringssystem ES über Typkennungen verwaltet und adressiert, so daß deren Verschaltung möglich ist.

Zur Verschaltung der Basisobjekte BO zu einer konkreten Steuerung ist das auf einem Personal Computer/Programmiergerät ablaufende Engineeringssystem ES vorgesehen, das Off-Line arbeitet und auf Abbilder sämtlicher Basisobjekte BO Zugriff hat. Im Einzelfall kann die Verschaltung der Basisobjekte BO alternativ auch über ein "internes"

Bootobjekt BOO vorgenommen werden.

Mit Hilfe des Engineeringssystems ES wird die konkrete Steuerungslösung programmiert und unter Einsatz entsprechender Werkzeuge in Betrieb genommen. Durch die Verschaltung und Konfiguration der Basisobjekte BO entstehen Applikationsmodule SAM mit bestimmter, den Hardwarebaugruppen entsprechender Funktionalität (Einachsmodul, Mehrachsmodul, etc.).

Die Ausführung einer konkret projektierten Softwarestruktur erfolgt im Runtime-System RS.

Die wichtigsten Basisobjekte BO, sind die Reglerobjekte FCO, die Führungsgrößenobjekte CVO und die Programmverarbeitungsobjekte PPO, wobei im folgenden diese Objekte, beginnend mit dem Programmverarbeitungsobjekt PPO, anhand von Fig. 2 näher dargestellt werden.

Ein Programmverarbeitungsobjekt PPO ist primär eine zur Abarbeitung eines Anwenderprogramms AP fähige virtuelle Maschine. Das Programmverarbeitungsobjekt PPO beinhaltet alle Funktionen zur Ausführung einer Befehlssequenz in einem zeitlichen Multi-Tasking-Kontext. Das jeweilige Anwenderprogramm AP ist in einer speziellen Assemblerkodierung MCASM der virtuellen Maschine abgelegt, nachdem es auf dem Engineeringssystem ES graphisch APG oder textuell APT erstellt, kompiliert und von diesem an das jeweilige Programmverarbeitungsobjekt PPO transferiert wurde. Der Zugang zu BO- und Betriebssystemfunktionalitäten wird somit über den Befehlssatz der virtuellen Maschine realisiert.

Für ein MC-Applikation werden die jeweiligen Steuerungsprogramme direkt in der Art mehrerer Tasks geschrieben und angelegt. Zur Abarbeitung des jeweiligen Steuerungsprogramms werden vom Programmverarbeitungsobjekt PPO Tasks AT, ATZ, ATS im Multi-Tasking-System angelegt. Diese Tasks können für verschiedene Abarbeitungsmodi eingerichtet werden, wobei z. B. zwischen zyklischer Abarbeitung – zyklische Task ATZ, – entsprechend der Arbeitsweise z. B. einer speicherprogrammierbaren Steuerung und einer sequentiellen Abarbeitung – sequentielle Task ATS – z. B. bei der Berechnung einer Bewegung unterschieden wird. Dabei können sich die Tasks selbstverständlich gegenseitig beeinflussen, indem z. B. aus einer zyklischen Task ATZ die Koordination einer sequentiellen Bewegungstask ATS erfolgt. Aus diesen Anwendertasks AT, ATZ, ATS heraus werden weitere Tasks über MCASM-Instruktionen zur Nutzung der Funktionen der Reglerobjekte FCO, der Führungsgrößenobjekte CVO und/oder der Treiberobjekte DRO angelegt.

Eine einzelne IAP stellt eine Anweisung auf MCASM-Niveau dar. Eine IAP kann PPO-interne und PPO-externe Schnittstellen von anderen BO's bzw. des Betriebssystems zur Realisierung seiner Funktion benutzen. Der Aufbau und die Verwaltung PPO-externer Verbindungen erfolgt durch den MNG. Das PPO startet zur Verarbeitung der geladenen IAP-Sequenz, die im ES aus dem AP erzeugt wurde, eine Default-ATS bei einer definierten IAP (entspricht Programm-Eintrittspunkt). Es existieren IAP's, die eine rekursive Task-Nutzung ermöglichen. Dazu sind im MCASM-Befehlssatz z. B. IAP's vorhanden, die das Erzeugen einer neuen ATS oder ATZ und deren Start ab einer bestimmten IAP innerhalb der IAP-Sequenz ermöglichen.

Neben Standardoperationen eines Rechners, Ein-/Ausgabefunktionen sowie speziellen Funktionen, die ansonsten z. B. von einer speicherprogrammierbaren Steuerung bereitgestellt würden, werden Funktionen zur Berechnung von Bewegungen und zur Regelung benötigt. Diese höheren Funktionen stellen z. B. die Führungsgrößenobjekte CVO und Reglerobjekte FCO zur Verfügung.

Ein Programmverarbeitungsobjekt PPO ist primär eine

zur Abarbeitung von IAP-Sequenzen fähige virtuelle Maschine. Eine einzelne IAP stellt eine Anweisung auf MCASM-Niveau dar. Sie bildet den programmierbaren Zugang des Anwenders zu den Funktionalitäten der BO's und des BS. Die Erzeugung und der Transfer der IAP-Sequenzen erfolgt in den Schritten:

- Erzeugung eines AP auf textuellem (APT) oder graphischem (APG) Wege im ES.
- Compilation des AP in IAP-Sequenzen (systemspezifische Kodierung; MCASM) der virtuellen Maschine des PPO im ES und Ablegen in einem Executable-File.
- Transfer der IAP-Sequenzen aus dem Executable an das jeweilige PPO durch das ES oder ein BOO.

Eine IAP kann PPO-interne und PPO-externe Schnittstellen von anderen BO's bzw. des BS zur Realisierung seiner Funktion benutzen. Der Aufbau und die Verwaltung dieser Verbindungen erfolgt durch den MNG.

Die über IAP's zugänglichen Funktionalitäten gliedern sich in:

- Anwender-Multitasking:
Dabei können neue Tasks mit zyklischer (ATZ) bzw. sequentieller (ATS) IAP-Bearbeitung auf der geladenen IAP-Sequenz erzeugt, zerstört und beeinflusst werden. Die dafür definierten IAP's benutzen dazu Funktionalitäten des BS.
- Verarbeitung von Standarddatentypen (Zahlen, Strukturierte Datentypen etc.)
- I/O-Operationen
- Zugang zu Spezialfunktionalitäten mit PPO verbundenen BO's wie z. B. CVO zur Bewegungsgenerierung oder FCO zur Regelung.

Der Start der Programmabarbeitung des PPO erfolgt auf Kommando des ES oder BOO über dessen Kommando-Schnittstelle. Dazu wird im PPO eine Default-ATS gestartet, die ihre Arbeit bei einer definierten IAP (entspricht Programm-Eintrittspunkt) der geladenen IAP-Sequenz beginnt.

Um physikalische Geräte M mittels eines Achs-Interfaces oder eines Ein-/Ausgabe-Interfaces über die konkrete Steuerungshardware des Systembausteins SB zu betreiben, werden Treiberobjekte DRO eingeordnet. DRO's dienen der Entkopplung von Objektschnittstellen von der konkreten Hardware, indem sie spezifische Details zu deren Ansteuerung implementieren. Im Ausführungsbeispiel kommt gemäß Fig. 2 zur Ansteuerung des Motors M ein Reglerobjekt FCO, ein Treiberobjekt DRO und ein Führungsgrößenobjekt CVO zum Einsatz. Der Ausgangswert des Reglerobjektes FCO dient über das Treiberobjekt DRO direkt zur Ansteuerung des Motors M. Das Führungsgrößenobjekt FCO, das in nicht dargestellter Weise mit dem Reglerobjekt FCO verschaltet ist, liefert zumindest die Führungsgröße für die z. B. mittels des Reglerobjektes FCO konstant gehaltene Drehzahl des Motors M. Des weiteren ist noch ein Treiberobjekt DRO für Eingabefunktionen E, mit denen sich z. B. Prozeßzustände erfassen lassen, und Ausgabefunktionen A, z. B. zum Einschalten bestimmter Prozeßkomponenten, vorgesehen.

Die Möglichkeiten hinsichtlich Konfiguration, Programmierung, Inbetriebnahme etc. werden nachfolgend anhand von Fig. 3 dargestellt.

Im Engineeringssystem ES wird eine konkrete Steuerungslösung entsprechend der jeweiligen Anforderungen des Kunden unter Einsatz entsprechender Werkzeuge VEW, KON, PRG (Verwaltung, Konfiguration, Programmierung) konfiguriert und programmiert, wobei die Inbetriebnahme

durch weitere Werkzeuge INB, MON, DEB (Inbetriebnahme, Monitoring, Debugging) unterstützt wird. Die Ausführung einer konkret projektierten Softwarestruktur mit dem zugehörigen Anwenderprogramm erfolgt im Runtime-system RS.

Das Engineeringsystem ES ermöglicht demzufolge die ingenieurmäßige Handhabung einer Bewegungssteuerungsapplikation (MC-Applikation) MCA und darüber hinaus auch die Repräsentation des Runtime-systems RS während des Engineerings (also von der Projektierung bis zur Inbetriebnahme).

Dazu hat das Engineeringsystem ES Zugriff auf ein Abbild sämtlicher im Runtime-system RS ausführbarer Basisobjekte BO. Demzufolge sind als Basisobjektklassen zumindest die oben bereits aufgeführten Reglerobjekte FCO, Führungsgrößenobjekte CVO, Programmverarbeitungsobjekte PPO, Treiberobjekte DRO und Systemmanagerobjekte SMO vorgesehen.

Sowohl im Engineeringsystem ES als auch im Runtime-system RS existiert über die Hardware HW (Programmiergerät oder Personal Computer HW1 für Engineeringsystem ES, Steuerungshardware HW2 für Runtime-system RS) ein vollständiges Laufzeitsystem mit Betriebssystem BS, Werkzeugen zum Systemmanagement und Basisobjekten BO.

Der Vorteil der vorliegenden Erfindung liegt vor allem darin, daß ein zu automatisierender technischer Prozeß mit erhöhter Funktionalität, z. B. im Hinblick auf Logik, Bewegung, Regelung und Messung, erhöhter Dynamik, insbesondere durch entsprechende Anpassung über eine geeignete Hardwareauswahl und flexibler Struktur, z. B. als Stand-Alone, zentrale oder dezentrale bzw. verteilte Lösung, handhabbar ist.

Für die logischen Abläufe, Bewegungsvorgänge oder Regelungen stehen Basiselemente oder Basisfunktionen zur Verfügung. Die Basiselemente sind dabei verbindbare Komponenten, die Basisobjekte BO, die die Grundkomponenten für Automatisierungsaufgaben mit jeweils einheitlichen Schnittstellen darstellen. Die Basisfunktionen stehen im Rahmen programmierbarer Funktionen zur Verfügung, wobei ein Grundbefehlssatz für Logik, Bewegung (Einzelachse, lose und enge Master-Slave-Kopplung, Geometrie-Verbund) und Regelung verwendbar ist. Diese Struktur ermöglicht eine freie Programmierbarkeit und eine flexible Anpassung an die Gegebenheiten des jeweiligen Automatisierungsvorhabens durch das Instanzieren und das Verbinden der jeweiligen Objekte.

Das Gesamtsystem stellt sich als Kombination aus einem Laufzeitsystem einerseits, mit einem Steuerungskern als verteilbarem Steuerungsbetriebssystem, und einem Engineeringsystem andererseits, das insbesondere die graphische Programmierung des Automatisierungsvorhabens über eine entsprechende Oberfläche ermöglicht, dar.

Diese Struktur kommt insbesondere einer natürlichen Herangehensweise bei der Automatisierung technischer Prozesse entgegen, nämlich einem schrittweisen Vorgehen, das mit der Festlegung der jeweiligen Funktionen beginnt, woraus sich logische Abhängigkeiten einzelner Funktionen oder Funktionsgruppen ergeben bzw. ableiten lassen und sich erst im Rahmen von Detailbetrachtungen mit konkreten Bewegungsvorgängen und dazu eventuell erforderlichen Regelungen befaßt.

Dies ermöglicht eine Einheit im Abarbeitungsprinzip, bietet gleichzeitig eine erhebliche Flexibilität bei der Systemgestaltung.

Eine nach diesem Ausführungsbeispiel konzipierte Steuerung kann beispielsweise angewandt werden für eine Verpackungsmaschine mit fünf Achsen im Hauptbereich, einem Temperaturregler und einem 3-Achs-Handlingsystem.

Im folgenden wird auf Fig. 6 verwiesen, in welcher eine Objektstruktur einer Schlauchbeutelmaschine gezeigt ist, wobei folgendes Vorgehen realisiert wird:

M1: Produkt-Zufuhrbahn

- Produktlänge konst. oder alternativ Produktabstand konst.
- Produktlängen- Toleranz... -20%; soll zur vergleichsweise gleichen Verpackungslängen-Toleranz führen und Verpackungsfolie sparen
- Produktabstände variabel (sowohl zu dicht als auch zu weit); soll die Verpackungslänge des Produkts nicht beeinflussen

M2: Längsnaht-Siegelstation (Master-Antrieb)

- Kontinuierliche Bewegung
- Folienabzug häufig mit über diesen Antrieb
- Arbeitswerkzeuge (geheizten Siegelrollen) werden bei Anlagenstillstand über SPS- Steuersignale außer Arbeitsstellung gebracht

M3: Quersiegelstation X- Richtung

- Bewegungsablauf:
 - Aufsynchonisieren auf Fixpunkt des Verpackungsstrangs (Mitte zwischen zwei Produkten)
 - Synchronlauf zum Fixpunkt
 - Absynchonisieren, Bewegungsumkehr
 - Rückbewegung
 - Zyklusende, Stillstand (nicht zwingend)
- Aufsynchonisierungspunkt über zuvor ermittelte Produktlänge und Formelwerk des Anwenders berechnet, im Anwenderprogramm hinterlegt und über Produkt-einstellungen parametrierbar.
- "Nebenbewegungen" (Absynchonisierkurve, Rückhub) an das Geschwindigkeitsniveau der Masterachse angepaßt

M4: Folienrollenantrieb

- Bewegungsanforderungen eines Wickelantriebs (Zentralwickler)
- Durchmessererfassung und Auswertung
- Einsatz insbesondere bei glatter Folie
- Steuermarken bei bedruckter Folie erfassen und Differenzen der Lieferung über Elastizität der Folie ausgleichen

M5: Quersiegelstation Y-Richtung

- Bewegungsablauf:
 - Schließen des Siegelwerkzeugs nach dynamisch optimierter Funktion (gegenwärtig häufig auftretende Beschleunigungs- und Bremsgeräusche sind unerwünscht)
 - Schließgeräusch der Siegelbacken ("Klatschen") sollen nicht auftreten, aber eine sichere Berührung ist erforderlich (Gleichnis "Pressen im letzten Bewegungsabschnitt" benutzt)
 - Dynamische Anpassung der Schließbewegung an die Elastizität und Lose der Schließmechanik erfordert Anpassung
 - Anpassung zeigt sich gegenwärtig stark Geschwindigkeits abhängig (neuer Arbeitspunkt neue Einstellung)
 - Zusätzliche Randparametern des Schließprofils Folienmitnahme (extra beschleunigungsbegrenzt)

- Verweilzeit im geschlossenen Zustand (konstant?)
- Schwellendes Moment über Drehmomentenregelung in der Verweilzeit
- Öffnung der Siegelbacken nach einer zweiten Funktion
- Zyklusende; Stillstand
- Bewegungshub der Siegelbacken von der Produkthöhe abhängig (Verstellung?, Meßeingang auswerten?)
- Verknüpfung der Bewegungsprofile X- und Y-Antrieb (Schließbewegung beginnt bereits vor dem Erreichen des Synchronpunkts??; Absynchronisieren beginnt vor vollständiger Öffnung??) ist noch zu betrachten
- An die Schließbewegung ist das Schneidsignal zu koppeln (dynamisch hochwertiger Ausgang; über SPS zu langsam).

Zusammenfassend läßt sich die Erfindung wie folgt kurz beschreiben:

Das Bewegungssteuerungssystem ist ein programmierbares, dezentral und flexibel aus verschiedenen Hardwarebaugruppen bestehendes Steuerungssystem, dessen Hardwarebaugruppen als Systembausteine bezeichnet werden.

Aufgrund der Vielschichtigkeit der bei der Bewegungssteuerung von Produktionsmaschinen zu erfüllenden Anforderungen wird von einem Sortiment an Hardwarebaugruppen, die mit ausreichender Rechenleistung, einem Echtzeitbetriebssystem und spezifischer Grundfunktionalität ausgerüstet sind, einem Netzwerk (z. B. Profibus) zum Aufbau eines dezentralen Systems mit verteilter Steuerungsfunktionalität sowie Bedien- und Beobachtungsgeräten mit vorprojektierten Oberflächen ausgegangen. Mit der Erfindung wird ein konfigurier-, verteil- und programmierbares Steuerungssystem zur individuellen Anpassung der Steuerungslösung an die Kundenanforderung vorgeschlagen, mit dem die projektierte Steuerungslösung auf Hardwarebaugruppen verteilt wird und wobei ein z. B. auf einem Programmiergerät oder auf einem Personal Computer ablaufendes Engineeringssystem zur Verwaltung, Konfiguration, Programmierung, zum Monitoring, Debugging und zur Inbetriebnahme eingesetzt wird.

Bezugszeichenliste:

AP Anwenderprogramm	45
APG graphisch erstelltes Anwenderprogramm	
APT textuell erstelltes Anwenderprogramm	
AT Applikationstask	
ATS sequentielle Applikationstask	
ATZ zyklische Applikationstask	50
BO Basisobjekt	
BOO Boot-Objekt	
BS Betriebssystem	
COM Kommunikation (Software)	
CVO Führungsgrößenobjekt	55
DEB Debugger	
DRO Treiberobjekt	
ES Entwicklungssystem	
FCO Reglerobjekt	
HW Hardware	60
HWn n-te Hardware	
IAP Instruktion im Anwenderprogramm (MCASM-Niveau)	
INB Inbetriebnahme-Tool	
KON Konfigurations-Tool	
MC Motion Control	65
MCA Motion Control Applikation	
MCASM Assemblersprache des MC-Systems	
MNG Manager	

MON Monitor	
Mx Motor x	
NW Netzwerk	
PC Personal Computer	
PPO Programmverarbeitungsobjekt	5
PRG Programmier-Tool	
RS Runtime-System	
SAM Einachs-Applikationsmodul	
SB Systembaustein	
SMO Systemmanagerobjekt	10
VEW Verwaltungs-Tool	

Patentansprüche

1. Verfahren zur Projektierung bzw. Programmierung einer Softwareapplikation zur Steuerung und/oder Überwachung eines externen technischen Prozesses, wobei instanziiierbare Basisobjekttypen (BO) mit adressierbaren Schnittstellen zur Parametrierung und Verschaltung vorgesehen sind, wobei als instanziiierbare Basisobjekttypen (BO) zumindest ein Programmverarbeitungsobjekttyp (PPO) zur Abarbeitung eines benutzerdefinierbaren Programms und zumindest ein Treiberobjekttyp (DRO) zur Ansteuerung der Prozeßhardware vorgesehen ist und wobei die Projektierung zumindest die Schritte der Selektion, Parametrierung und Verschaltung der Basisobjekte (BO) umfaßt.
2. Verfahren nach Anspruch 1, dadurch gekennzeichnet, daß als instanziiierbarer Basisobjekttyp (BO) ferner zumindest ein Führungsgrößenobjekttyp (CVO) zur Vorgabe einer Führungsgröße für eine Teilkomponente des externen technischen Prozesses vorgesehen ist.
3. Verfahren nach Anspruch 1 oder 2, dadurch gekennzeichnet, daß als instanziiierbarer Basisobjekttyp (BO) ferner zumindest ein Reglerobjekt (FCO) zur Regelung einer Teilkomponente des externen technischen Prozesses vorgesehen ist.
4. Verfahren nach einem der obigen Ansprüche, dadurch gekennzeichnet, daß die Projektierung auf einem Engineering-System (ES) erfolgt und daß die Software-Applikation auf einem Runtime-System (RS) zur Ausführung kommt.
5. Verfahren nach Anspruch 4, dadurch gekennzeichnet, daß sämtliche Basisobjekttypen (BO) im Engineering-System (ES) ein Abbild haben.
6. Verfahren zur Steuerung und/oder Überwachung externer technischer Prozesse, dadurch gekennzeichnet, daß die Steuerung und/oder Überwachung des technischen Prozesses mittels einer mit einem Verfahren nach einem der obigen Ansprüche projektierten bzw. programmierten Software-Applikation erfolgt.

Hierzu 6 Seite(n) Zeichnungen

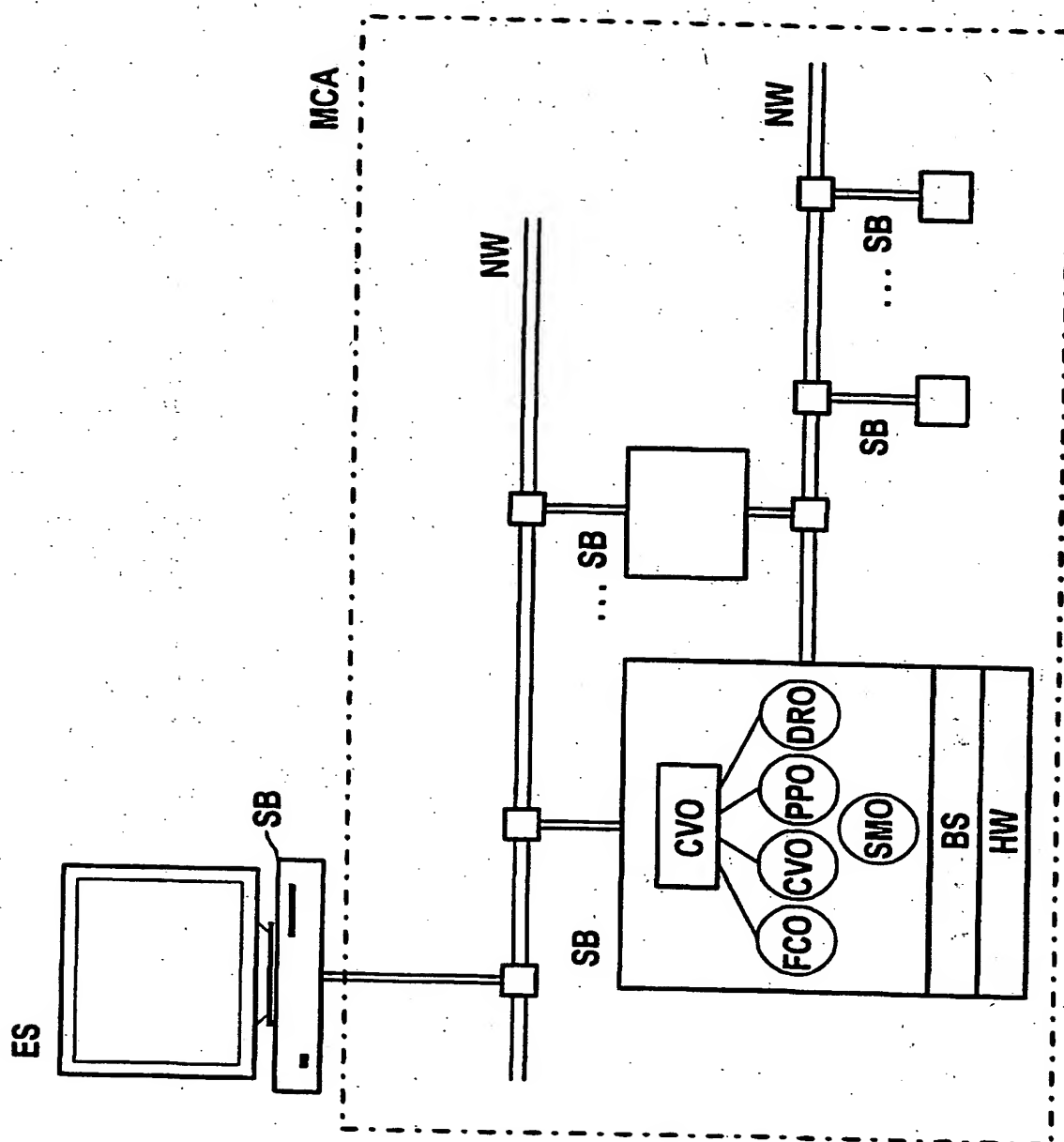


FIG. 1

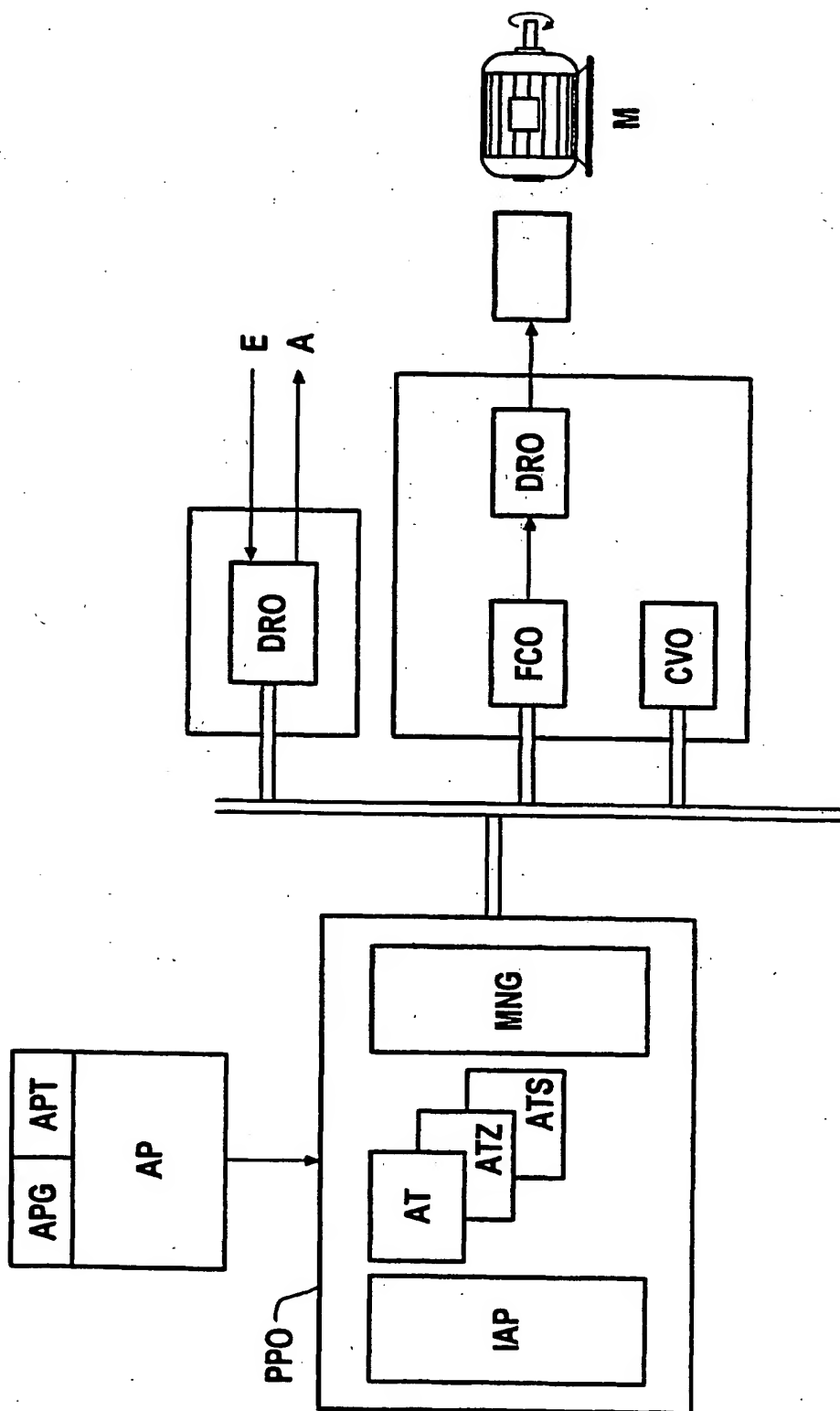


FIG 2

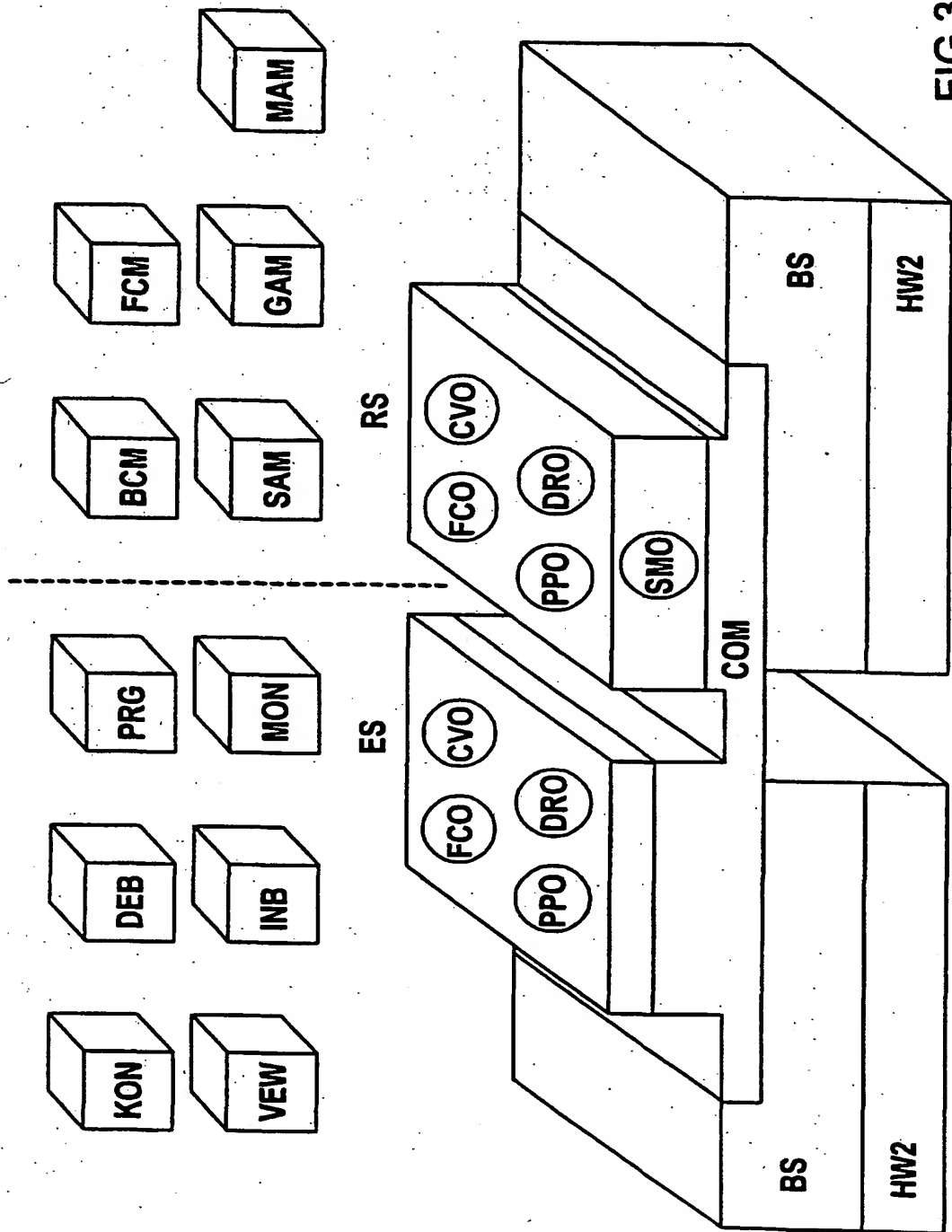


FIG 3

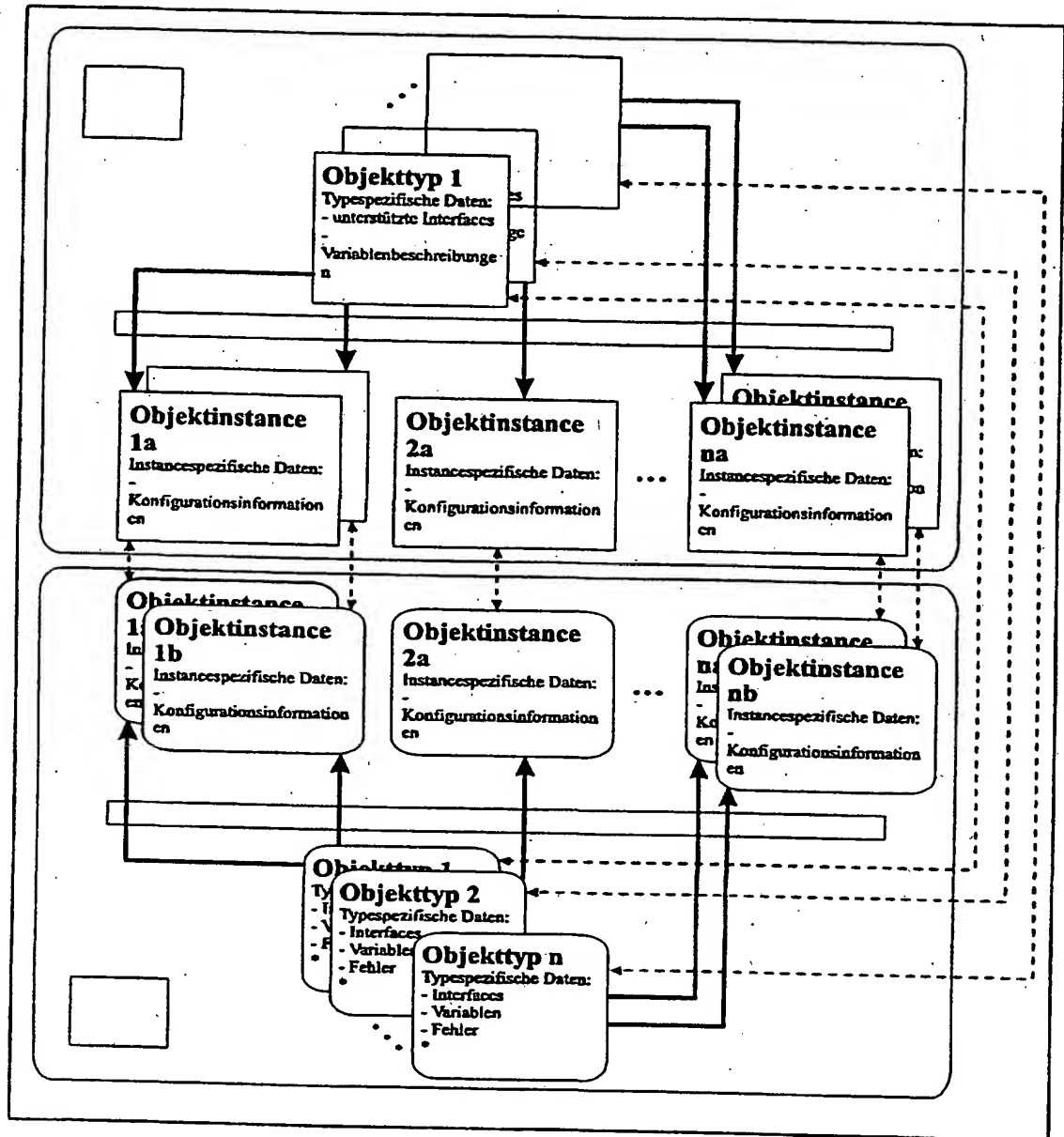


FIG 4

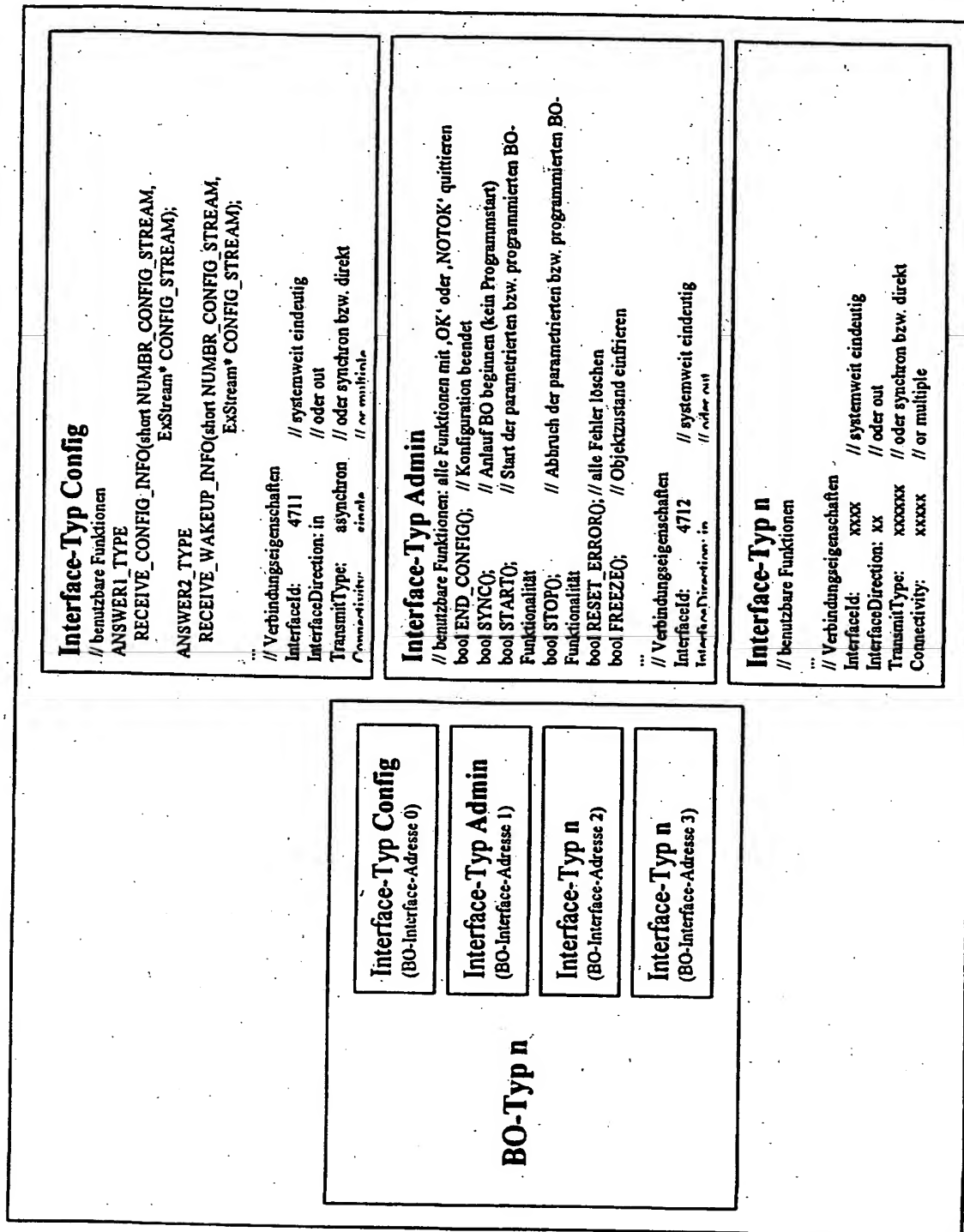


FIG 5

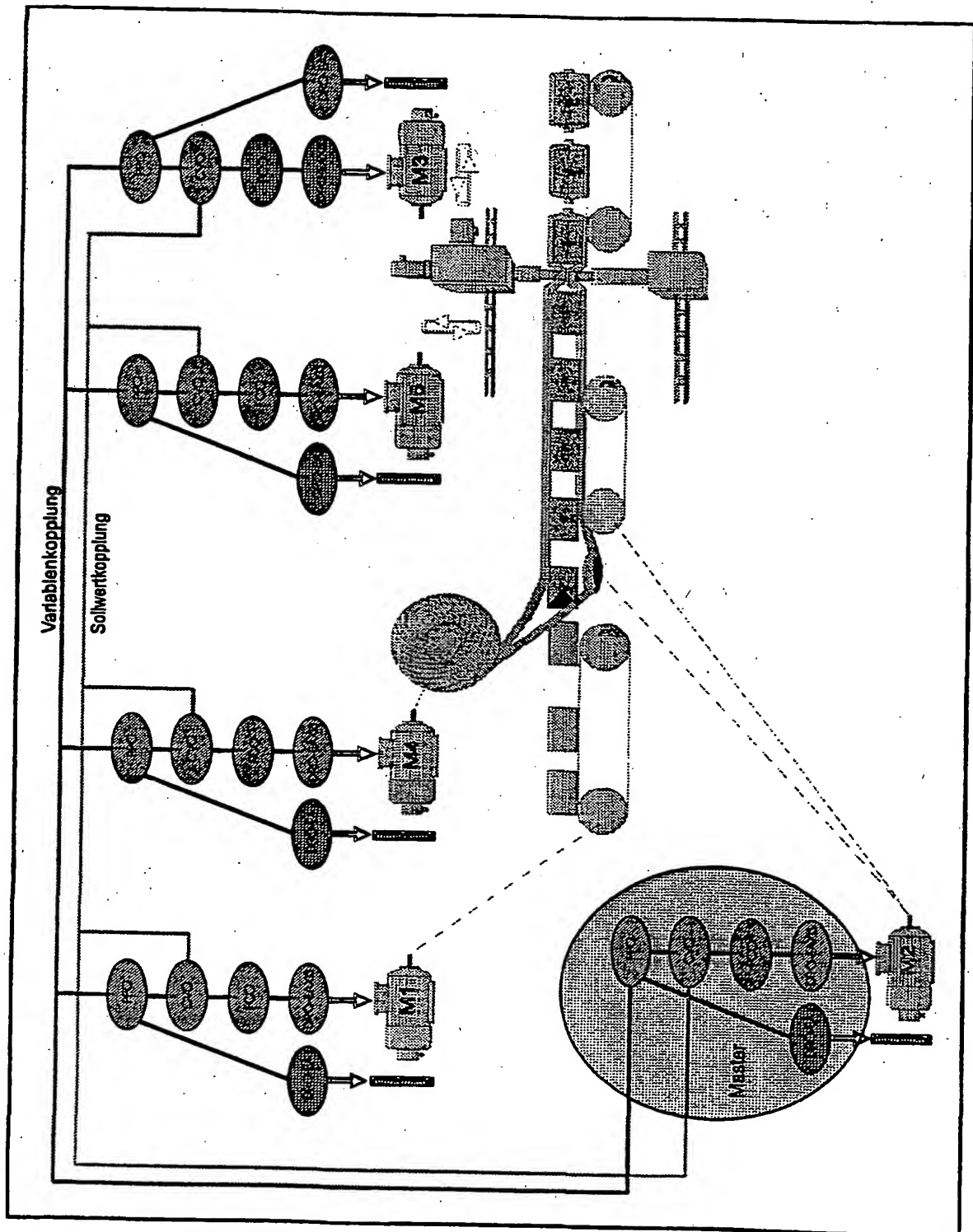


FIG 6

**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ BLACK BORDERS
- ☐ IMAGE CUT OFF AT TOP, BOTTOM OR SIDES
- ☐ FADED TEXT OR DRAWING
- ☒ BLURRED OR ILLEGIBLE TEXT OR DRAWING
- ☒ SKEWED/SLANTED IMAGES
- ☐ COLOR OR BLACK AND WHITE PHOTOGRAPHS
- ☐ GRAY SCALE DOCUMENTS
- ☒ LINES OR MARKS ON ORIGINAL DOCUMENT
- ☐ REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY
- ☐ OTHER: _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.

THIS PAGE BLANK